APPENDIX B

RADGUNS V.1.8 HIGH-LEVEL FLOW DIAGRAM

INTRODUCTION

The *RADGUNS* v.1.8 algorithm has been diagrammed at a high level (Appendix B). The diagram spans 26 pages in contrast to the low-level CASE tool output from the Phase 1 *RADGUNS V.1.7* verification analysis, which filled numerous large notebooks.

A Flow Chart Key has also been included to explain the meaning of the symbols used in the diagram. Each block in the diagram is numbered and a discussion of each block comprises the contents of the Block Description section.

METHODOLOGY AND APPROACH

To develop a high-level flow diagram, two engineering roles of expertise are required. The electrical engineer role specifies the functional element algorithms of the model while the software engineer role finds the algorithms in the code, analyzes the CASE tool output and maps the high-level diagram. These two roles could be performed by one person, and can even overlap over time, but for purposes of this report these roles will be considered as stages. This report will only discuss the software engineer's contribution to the diagram process.

A methodology for creating a high-level block diagram such as the one in this report requires a more complex set of design criteria than would be necessary for the creation of a low-level diagram where each line of code is diagrammed. The preliminary step to creating the high-level diagram is to perform the same CASE tool analysis on the code as was done for the Phase 1 *RADGUNS V.1.7* verification delivery. The CASE tool output information is described below:

a. Structure Chart - This chart pictorially maps the calling and called routines for the entire program in one chart. The software engineer uses this chart to get an overview of the program being verified. Because of the visual nature of this chart, the software engineer can instantly understand the calling hierarchy of the analyzed program, which saves time leafing through pages of code. The name of the CASE program that generates this output is called: *AutoFlow-FORTRAN*, by AutoCASE Technology.

b. Nesting Level/Cross Reference Table - Indicates the nesting level of each routine by drawing a line in the left hand margin of a source code file which spans all the code lines in a particular nesting level. This output also includes a cross reference table of variables used in a particular routine and denotes which lines the variables occur on. Both the cross reference table and the nesting level diagram lines help the software engineer to quickly detect the variables and associated code for a particular algorithm. The name of the CASE program that generates this output is called: *Source Print* by Powerline.

c. Low-Level Block Diagram - Creates a line by line low-level flow diagram for each subroutine. The software engineer uses these diagrams to analyze complicated algorithms, but will not use these diagrams for the simpler subroutines. The name of the CASE program that generates this output is called: *AutoFlow-FORTRAN,* by AutoCASE Technology.

d. Tree Diagram - Indicates the number of times a particular subroutine is called in the program and whether that call was recursive. It also indicates the name of the source file this routine resides in, and lists the routines that it calls. the software engineer uses this output, along with the structure chart mentioned above to determine quickly the calling hierarchy of the program being verified. The name of the CASE program that generates this output is called: *Tree Diagrammer*, by Powerline.

The software engineer uses the CASE tool output along with a copy of the source code to study the algorithms in the program being verified. The software engineer creates a preliminary copy of the high-level flow diagram. The electrical engineer looks at the diagram and determines if the algorithms pulled out of the code are correct. If not, the software engineer returns to the code to find the missing component of a particular algorithm, and resubmits the updated diagram to the electrical engineer. It is necessary to incorporate this iterative process between the software and electrical engineer, as the task of verifying a program is a team process.

DIAGRAM COMPONENTS

The software engineer then studies these outputs in preparation for creating the high-level diagram. The diagram is going to consist of 3 types of code entities:

1. Functional Element (FE) - A logical component of the model, for example the antenna of a radar. An FE can encompass one or several routines.

2. Code block - A section of code that performs a significant function in the model but is not an FE. A code block may be composed of one or several routines. An example of a code block in *RADGUNS* v.1.8 is MPATH1 (block 26) which computes the multipath effect calculations that do not change on a pulse to pulse basis.

3. Toolbox Routines - These are little helper routines that are used frequently and perform utility services. An example of toolbox routines in *RADGUNS* v.1.8 is PAGE (block 7) which determines the page layout.

Note: Sometimes there may be a fine line in the criteria used to distinguish between a code block and a toolbox routine. Since the goal of a high level flow diagram is to facilitate ease of reading and clarity, an occasional border line defined entity (between a code block or toolbox routine) is not in danger of degrading the utility of the diagram.

HIGH-LEVEL DIAGRAM

After the CASE tool information has been gathered and studied, the following guidelines are then used to create the high-level diagram:

a. Block contents: FEs and code blocks were represented by one block in the diagram. If an FE or a code block called either an FE or another code block, these called entities were also represented by a block in the flow diagram. Toolbox routines were lumped together in the same block as the routine they

were called by. Using this rule of thumb, the structure chart was reduced down to a diagram with 99 blocks.

b. Decision blocks: To keep the diagram high-level and simplified, decision blocks were only used to link one of the three code entities listed above. Decision blocks to an entity smaller then these were not diagrammed (i.e. a decision block to a line of code that sets a factor to a certain value would not be diagrammed).

c. Function emphasis: Since the critical function of *RADGUNS* v.1.8 is to simulate aircraft survivability data, code that performs software housekeeping functions was given less detail in the diagram than code that performs engineering functions. For example, 4 routines were used to initialize data in *RADGUNS* v.1.8. In the high-level flow diagram this operation was represented by one block (block 1).

d. Similar routines together in one block: When ever routines that performed a similar function were found in the same proximity of the code, they were put together in one block. For instance, PAGE, TOPBOT, and HEADER were grouped into one block since they all perform page layout and printout functions.

e. Format: The format that the diagram is drawn in makes a big difference in facilitating the reader's ability to quickly understand the contents. These are some format conventions that were adhered to:

1. Symbols: Conventional flow chart symbols were used where ever they were appropriate. All symbols used are explained in the Flow Diagram Key in Appendix B.

2. One nesting level per page: When ever possible, one nesting level was placed on one page (or across multiple pages). Off-page connectors were used to represent large blocks of code that were depicted on other pages.

CONCLUSION

The goal of this flow diagram, was to aid the technical and administrative person in making a quick assessment of the contents of *RADGUNS* v.1.8. The level of depth depicted in the diagram and the guidelines chosen to create this diagram were carefully considered.

High Level Block Diagram Description

Block 1: GUNDAT, RDRDAT, WPNDAT, and GENDAT initializes the gun, radar, weapon, and other variables.

Block 2: Initializes either the radar cross section-plot (RCSPLT) or the "detection range only" (DETRNG) simulation for plotting.

Block 3: RCSPLT generates a file containing RCS data for any chosen target modeled during a 'RCSP' simulation type.

Block 4: TXMTR (in RAD5 only) models radar transmitter function.

Block 5: EVENT (in RAD5 only) records the occurrences of important events within the scenario. The events are printed in chronological order in the final report. PAGE determines the number of lines per page and the need for page breaks. TOPBOT writes the classification at the top and bottom of the page, and HEADER writes the header at the top of the page.

Block 6: RDRINI initializes several radar system parameters for the acquisition and track radar.

Block 7: EVENT records the occurrences of important events within the scenario. The events are printed in chronological order in the final report. PAGE determines the number of lines per page and the need for page breaks. TOPBOT writes the classification at the top and bottom of the page, and HEADER writes the header at the top of the page.

Block 8: The search mode decision block chooses between a sector search pattern (SRCH1) in RAD1 only, circular search pattern (SRCH2), and cueing the antenna directly at the target (PERCUE).

Block 9: WRTMUL creates files of data for plots.

Block 10: Determines page layout as specified by PAGE, TOPBOT, and HEADER (see Block 7).

Block 11: REPORT prints the events in a chronological order in a final report format with a page layout specified by PAGE, TOPBOT, and HEADER (see Block 7).

Block 12: INPUT1 allows the user to enter weapon and scenario-defining system inputs into the program and prints the information in the final report.

Block 13: RDCMNT reads comment lines in the input parameters file. The input is printed to the final report with a page layout specified by PAGE, TOPBOT, and HEADER (see Block 7).

Block 14: CHKTRK checks the weapon system track type to ensure it is realistic and prints out warning messages for unvalidated or not-yet-developed track modes with a page layout specified by PAGE, TOPBOT, and HEADER (see Block 7).

Block 15: INPUT2 allows the user to enter weapon and scenario-defining system inputs into the program and prints the information in the final report.

Block 16: RDCMNT reads comment lines in the input parameters file. The input is printed to the final report with a page layout specified by PAGE, TOPBOT, and HEADER (see Block 7).

Block 17: DETRNG writes files of data for a "detection range only" simulation for plotting.

Block 18: Writes target altitude, range and elevation with a page layout specified by PAGE, TOPBOT, and HEADER (see Block 7).

Block 19: MOVTAR computes the target's position, velocity, and acceleration as a function of time.

Block 20: RDCMNT, BLUINP, BLUFPR, CONVTM, and BLUINT input the Blue Max flight path parameters from a disk file used to calculate the target position, velocity, and acceleration.

Block 21: INCTAR increments the initial target position for multiple run simulations.

Block 22: Writes target information with a page layout specified by PAGE, TOPBOT, and HEADER (see Block 7).

Block 23: See Block 7.

Block 24: Blocks 24-28 are found in RAD1 only. SRCH1 performs a sector and circular search for the ZSU-23-4 radar system. It simulates the operators attempt to spot the target on the radar scope. When the target power divided by the clutter power and noise exceed a threshold, the operator spots the target on the PPI scope.

Block 25: ORIENT, and ROTATE compute the target's angular orientation with respect to the radar coordinate frame as a function of time.

Block 26: MPATH1 calculates those multipath effect calculations that do not change on a pulse-by-pulse basis.

Block 27: See Block 7.

Block 28: PDET, and PRBDET calculate the probability of detection of the target.

Block 29: In SRCH2 the radar searches the sky for a target in a circular scan mode. It simulates the operators attempt to spot the target on the radar scope. When the target power divided by the clutter power and noise exceed a threshold, the operator spots the target on the PPI scope.

Block 30: See Block 4 (in RAD5 only).

Block 31: See Block 5 (in RAD5 only).

Block 32: See Block 25.

Block 33: See Block 7.

Block 34: See Block 26.

Block 35: See Block 28.

Block 36: PERCUE returns the time the target would be detectable if the antenna was perfectly aimed at the target.

Block 37: See Block 25.

Block 38: See Block 26.

Block 39: See Block 7.

Block 40: See Block 28.

Block 41: ENGAGE controls the weapon system while in autotrack mode. It tracks the target, sends target coordinates to the fire control computer, aims guns, and fires at the target.

Block 42: See Block 7.

Block 43: See Block 4 (in RAD5 only).

Block 44: See Block 5 (in RAD5 only).

Block 45: See Block 25.

Block 46: FILROT, and CONSCA models a beam that nutuates about the antenna boresight maintaining a constant angle during a conical scan.

Block 47: SIGNL, and MPATH1 calculate the target power (including multipath effects) and ground clutter return power.

Block 48: MOVJAM computes the position and velocity of all active jammers.

Block 49: SIGJAM calculates the gain of the radar transmit antenna in the direction of any active jammers.

Block 50: REACQR computes boresight azimuth, elevation, and range during manual reacquisition of the target following a breaklock. The operator error while trying to move the antenna back onto the target is simulated.

Block 51: See Block 7.

Block 52: OPTMCS simulates the operation of an optical mechanical computing sight. It generates lead angles for the guns based on estimates of target range, speed, and approach angle. SPDRNG simulates the operation of speed rings and returns the gun position. Both subroutines call OPTRAK which models the operator attempt to keep the cross-hairs of a telescope aligned with the target.

Block 53: BARFIR aims the guns in a user-specified direction for barrage fire.

Block 54: CHGTRK changes tracking modes in response to a breaklock caused by jamming.

Block 55: See Block 14.

Block 56: See Block 9.

Block 57: See Block 10.

Block 58: RCVRT is the radar receiver model that processes each echo signal from the target, calculating angle and range channel outputs.

Block 59: See Block 7.

Block 60: PHASE (in RAD1 and RAD4 only) is the phase detector in the radar receiver. It calculates the azimuth and elevation error signals which are used to drive the antenna servos.

Block 61: PHASE (in RAD2 only) is the phase detector in the radar receiver. It calculates the azimuth and elevation error signals which are used to drive the antenna servos.

Block 62: PHIDEL computes the phi and del matrices needed for the time domain solution of a differential equation.

Block 63: DERVIS calculates derivatives of a function using differences between successive function values.

Block 64: TIMDOM calculates the step-by-step time domain solution of a differential equation.

Block 65: MOVANT simulates the antenna servo system, moving the antenna in response to error signals from the receiver.

Block 66: See Block 62.

Block 67: See Block 63.

Block 68: See Block 64.

Block 69: RSERVO calculates the new radar range to the target by simulating the range tracking servos.

Block 70: See Block 62.

Block 71: See Block 63.

Block 72: See Block 64.

Block 73: JAMMER simulates up to ten simultaneous jamming techniques. Generates instantaneous jamming signal level as a function of time for various types of jamming.

Block 74: NOISE simulates a barrage noise jammer whose signal may be amplitude modulated.

Block 75: INVERS simulates an inverse gain noise jammer whose signal may be amplitude modulated.

Block 76: REFLECT simulates the effect of a passive reflector.

Block 77: RGWO simulates a range-gate walkoff jammer whose signal may be amplitude modulated.

Block 78: RPTR simulates a straight-through repeater jammer whose signal may be amplitude modulated.

Block 79: SWEPTA simulates a swept audio noise jammer whose signal may be amplitude modulated. The jammer on-off frequency is linearly swept across a user specified frequency range.

Block 80: See Block 7.

Block 81: FIRCON simulates fire-control computer. It estimates the target flight path and calculates the direction that the guns should be aimed to hit the target, taking the vertical drop of the shell into account.

Block 82: FIL2AB fills two matrices with the coefficients for the target position predictor in the second order fire-control computer.

Block 83: See Block 62.

Block 84: FILLAB fills two matrices with the coefficients for the target position predictor in the fire control computer.

Block 85: See Block 64.

Block 86: MOVGUN simulates the gun servo system which moves the guns in response to the coordinates sent by the fire-control computer.

Block 87: See Block 62.

Block 88: See Block 63.

Block 89: See Block 64.

Block 90: SHOOT synchronizes the firing of the guns with the rest of the simulation and checks the status of the gun and fire light to determine the proper action to take.

Block 91: BURST schedules a burst of fire and determines the mandatory rest period for the gun barrels to cool off.

Block 92: See Block 7.

Block 93: HITPRB calculates the probability of hitting the target for a shell fired during the current time interval and updates the cumulative probability of hit for all the shells fired so far.

Block 94: See Block 25.

Block 95: INTGTA computes the presented area of the target for a given elevation and azimuth from the target to the shell.

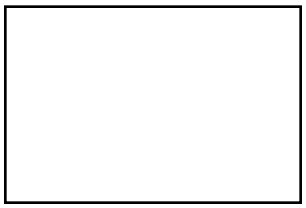
Block 96: See Block 7.

Block 97: See Block 9.

Block 98: See Block 10.

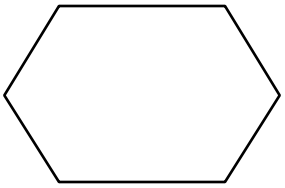
Block 99: SALVO computes the probability of hit/kill for the presented area and current range using the given weapon firing characteristics.

Flow chart key



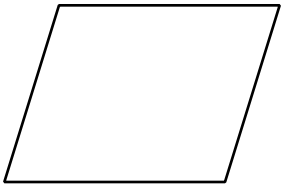
Indicates a process, or a logical componant of the code.

Offpage connector- represents a section of code that can be found on another page. When that section terminates the point.

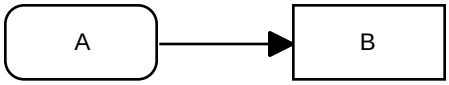


Indicates a logical component of the code that performs housekeeping such as constant initiallization

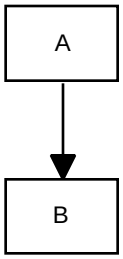
Terminal - Indicates start or stop point of a section of code



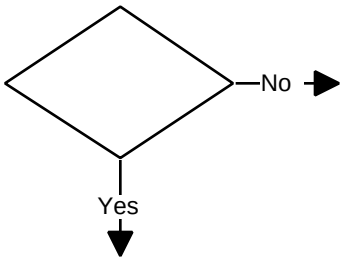
Input/Output - Indicates logical component that performs I/O



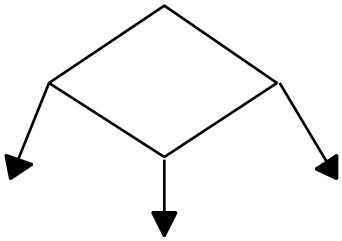
A calls B, and the routines associated with B



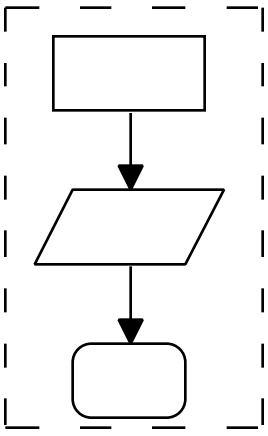
B is executed immediately after A in the flow sequence



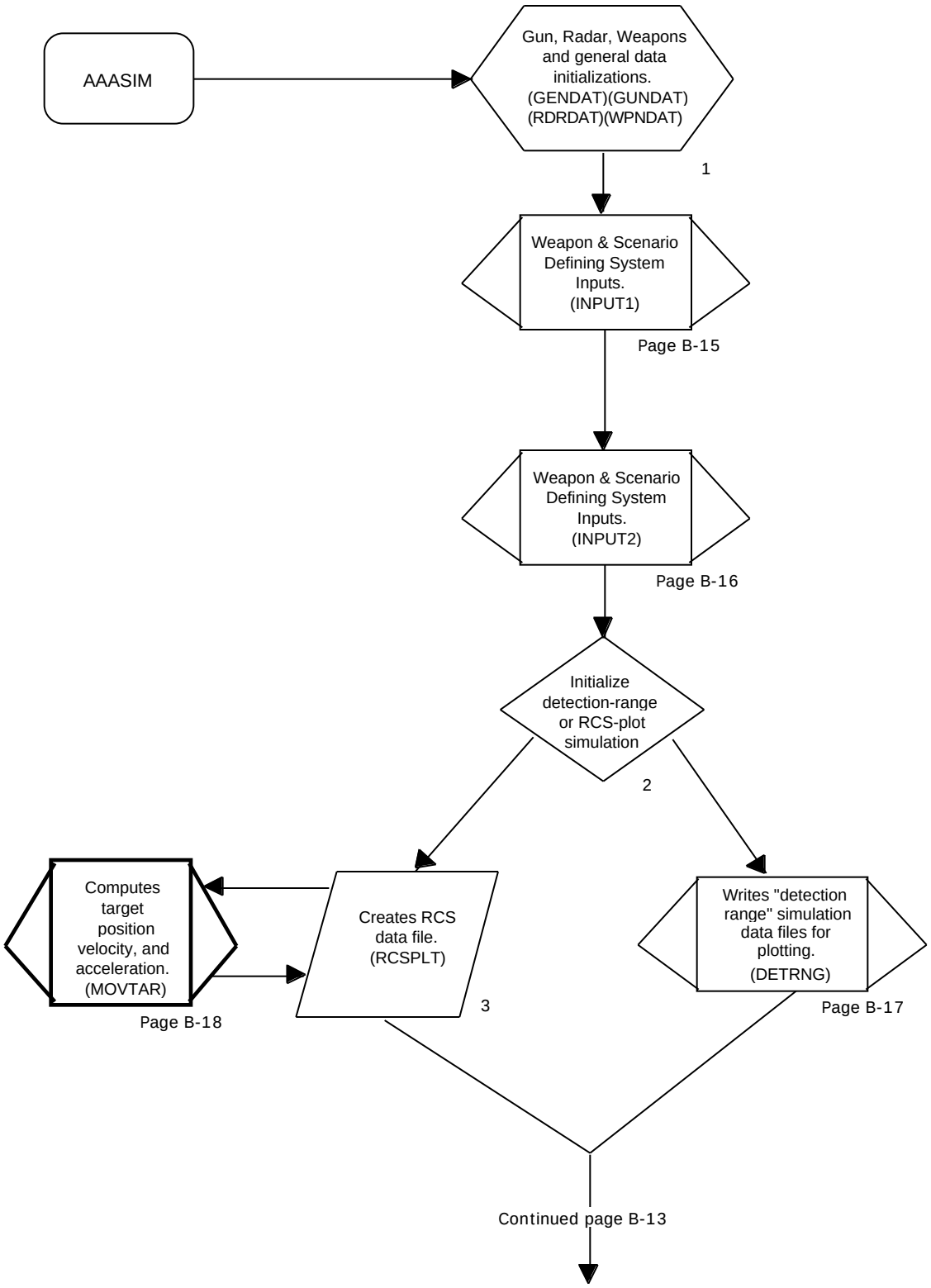
Decision Block

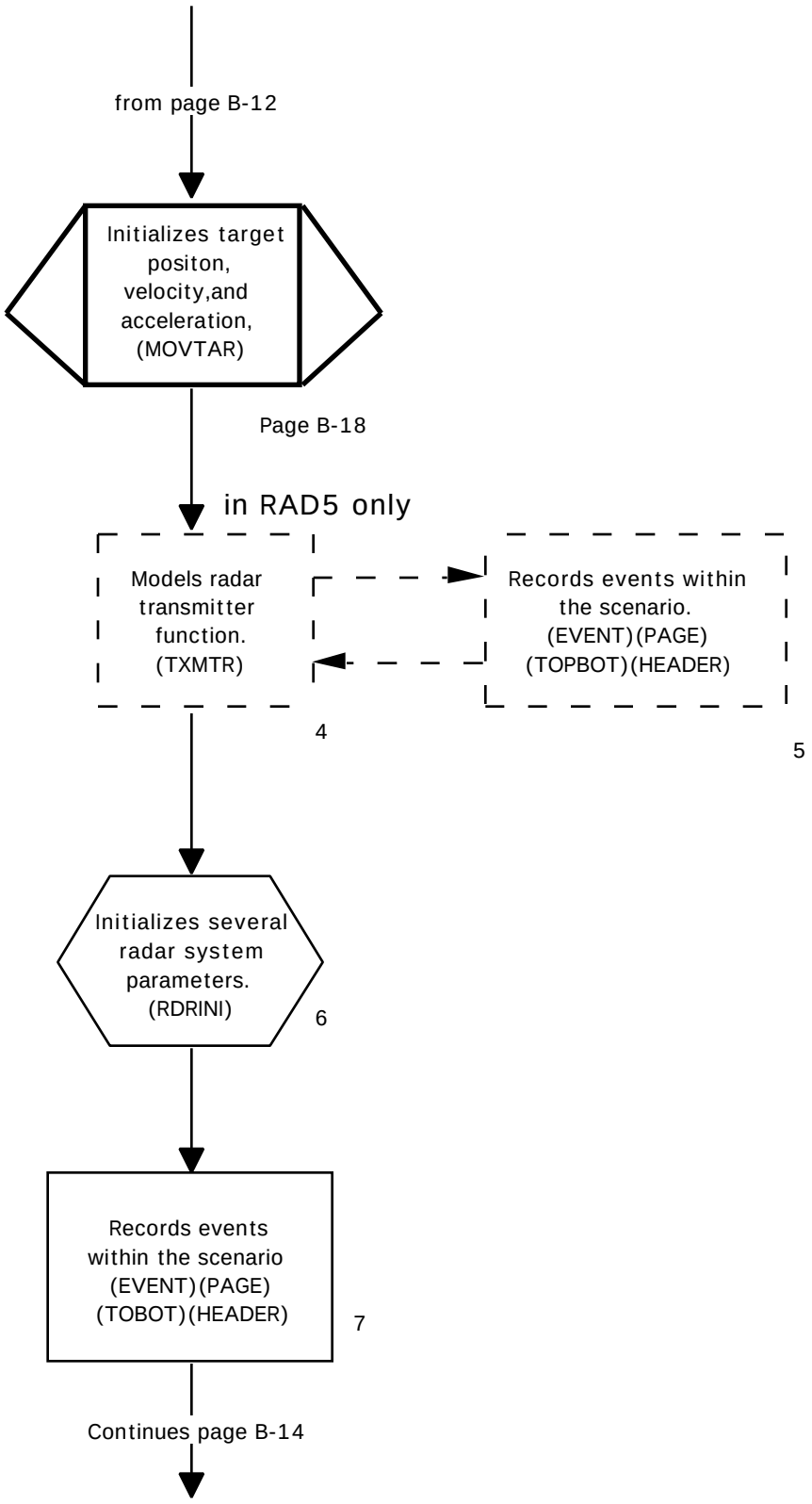


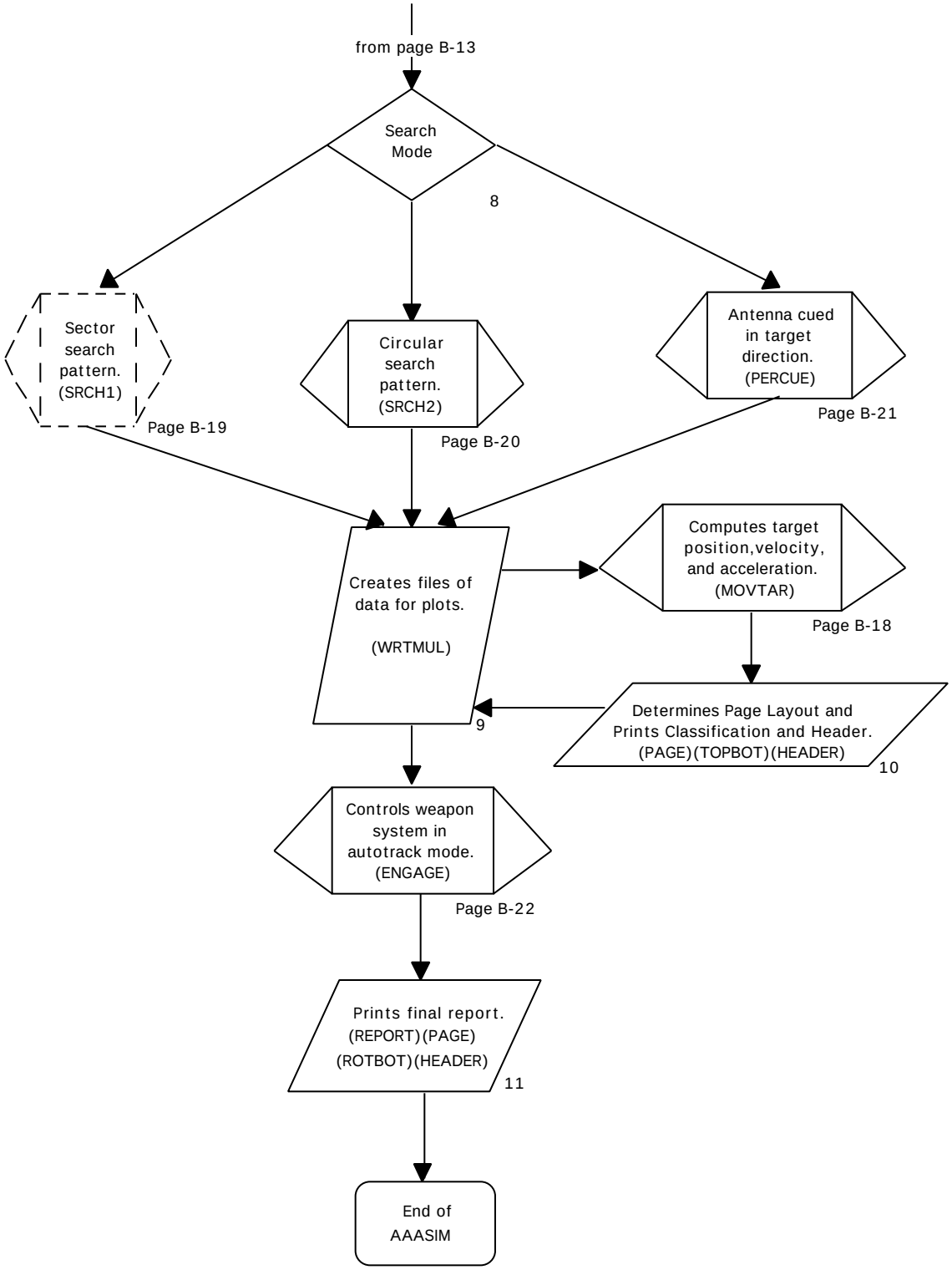
High level decision box simulates a CASE statement and replaces many low level decision blocks

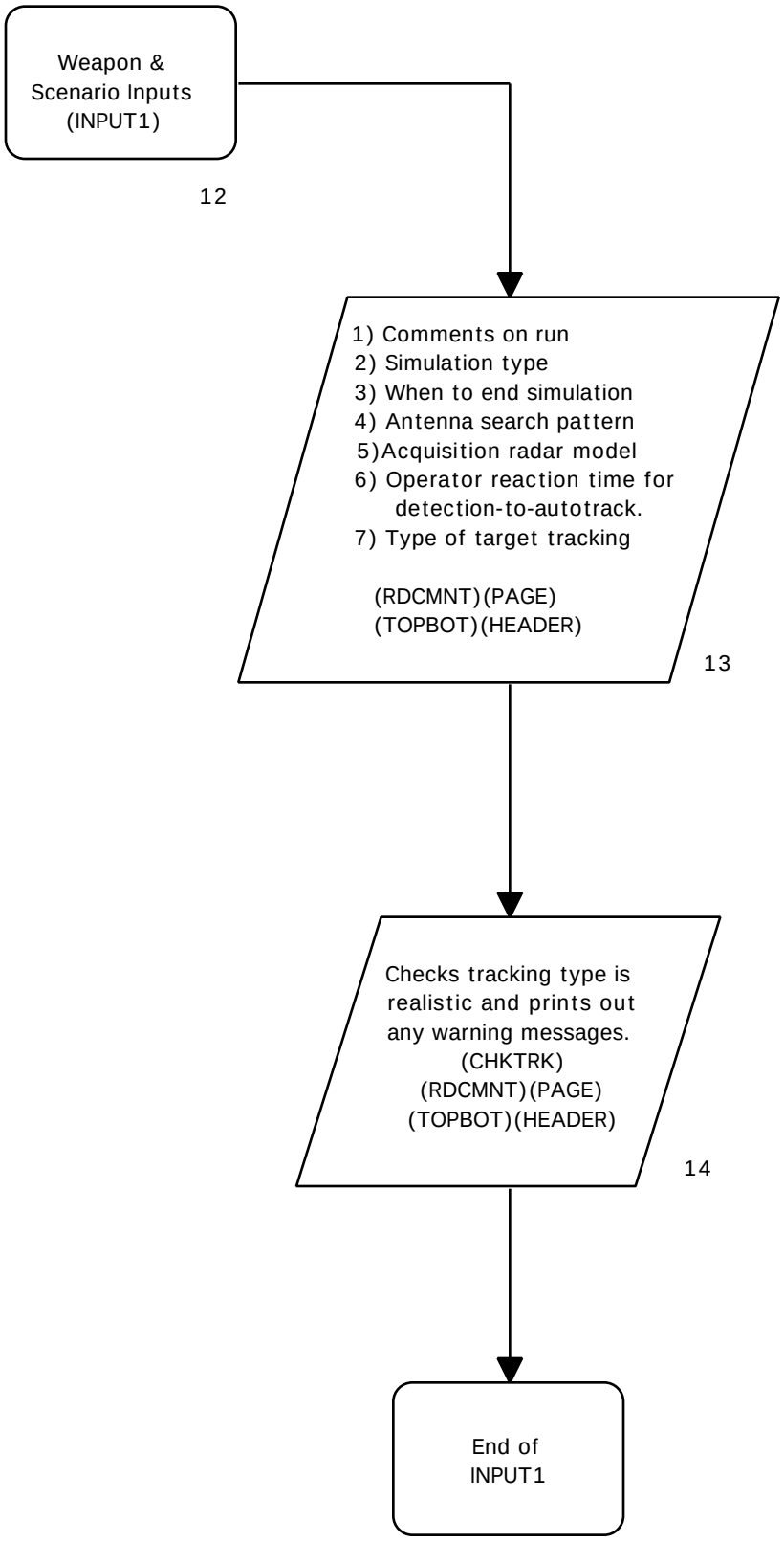


The outer box groups routines at one nesting level

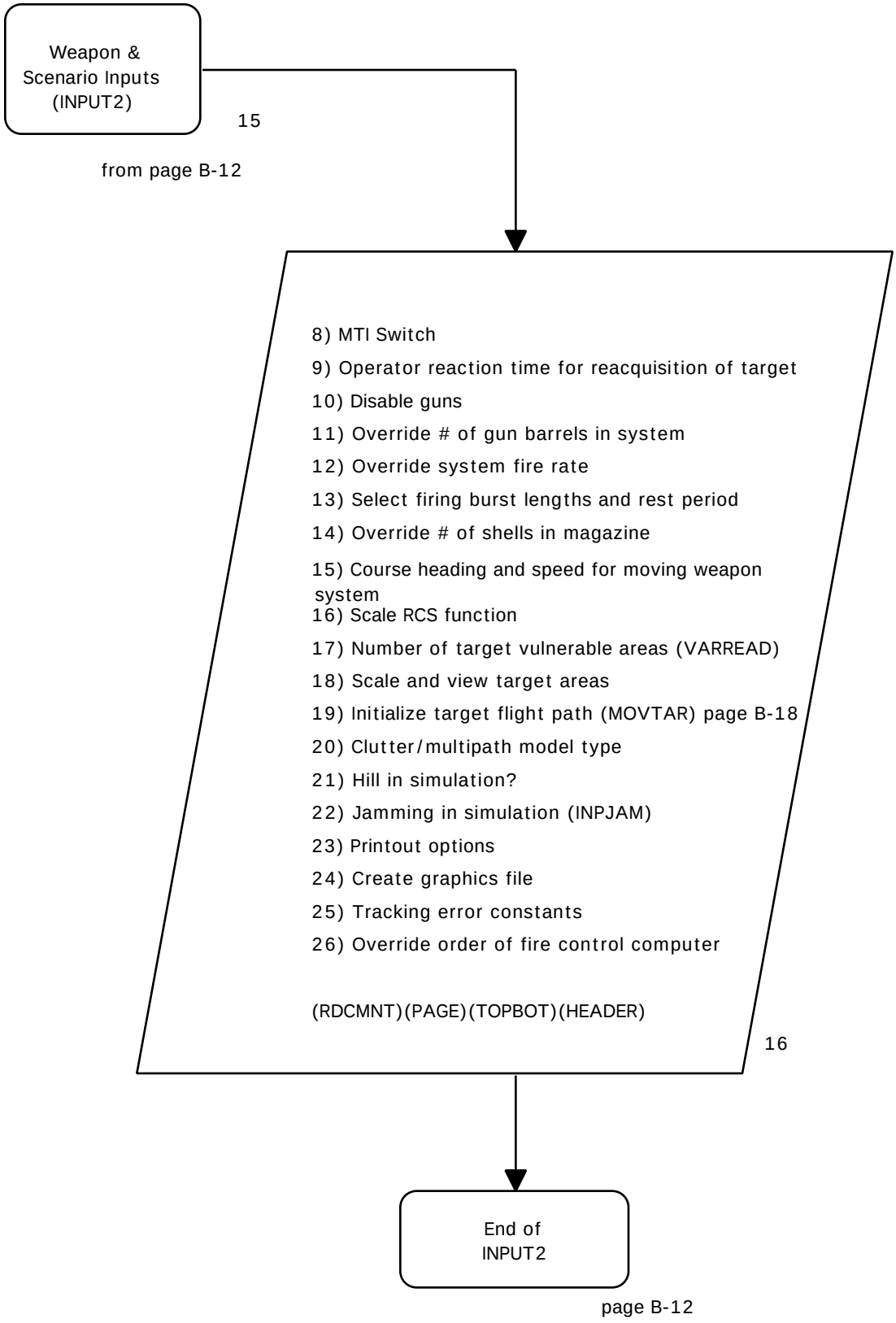

AAASIM
Gun, Radar, Weapons and general data initializations. (GENDAT)(GUNDAT) (RDRDAT)(WPNDAT)
1
Weapon & Scenario Defining System Inputs. (INPUT1)
Page B-15
Weapon & Scenario Defining System Inputs. (INPUT2)
Page B-16
Initialize detection-range or RCS-plot simulation
2
Computes target position velocity, and acceleration. (MOVTAR)
Page B-18
Creates RCS data file. (RCSPLT)
3
Writes "detection range" simulation data files for plotting. (DETRNG)
Page B-17
Continued page B-13

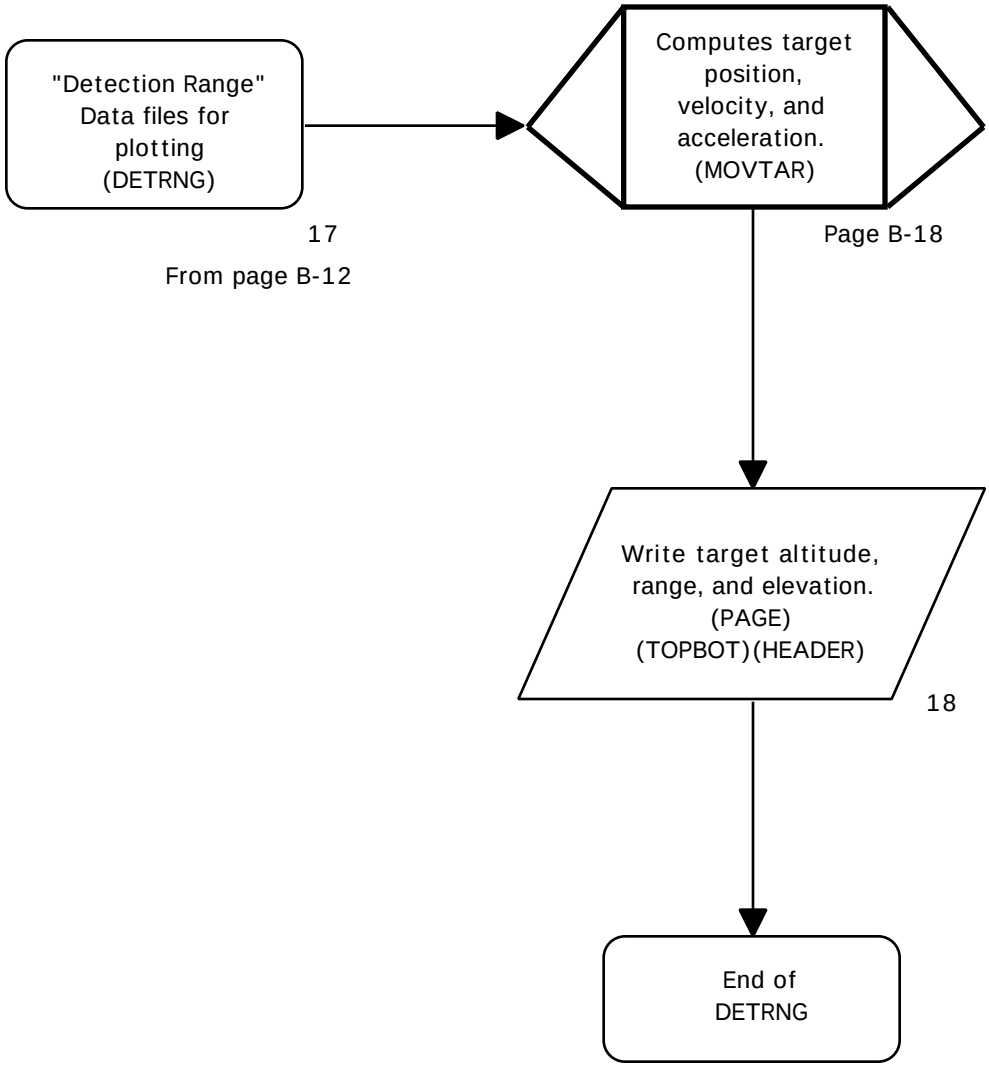

from page B-12
Initializes target positon, velocity,and acceleration, (MOVTAR)
Page B-18
in RAD5 only
Models radar transmitter function. (TXMTR)
4
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
5
Initializes several radar system parameters. (RDRINI)
6
Records events within the scenario (EVENT)(PAGE) (TOBOT)(HEADER)
7
Continues page B-14

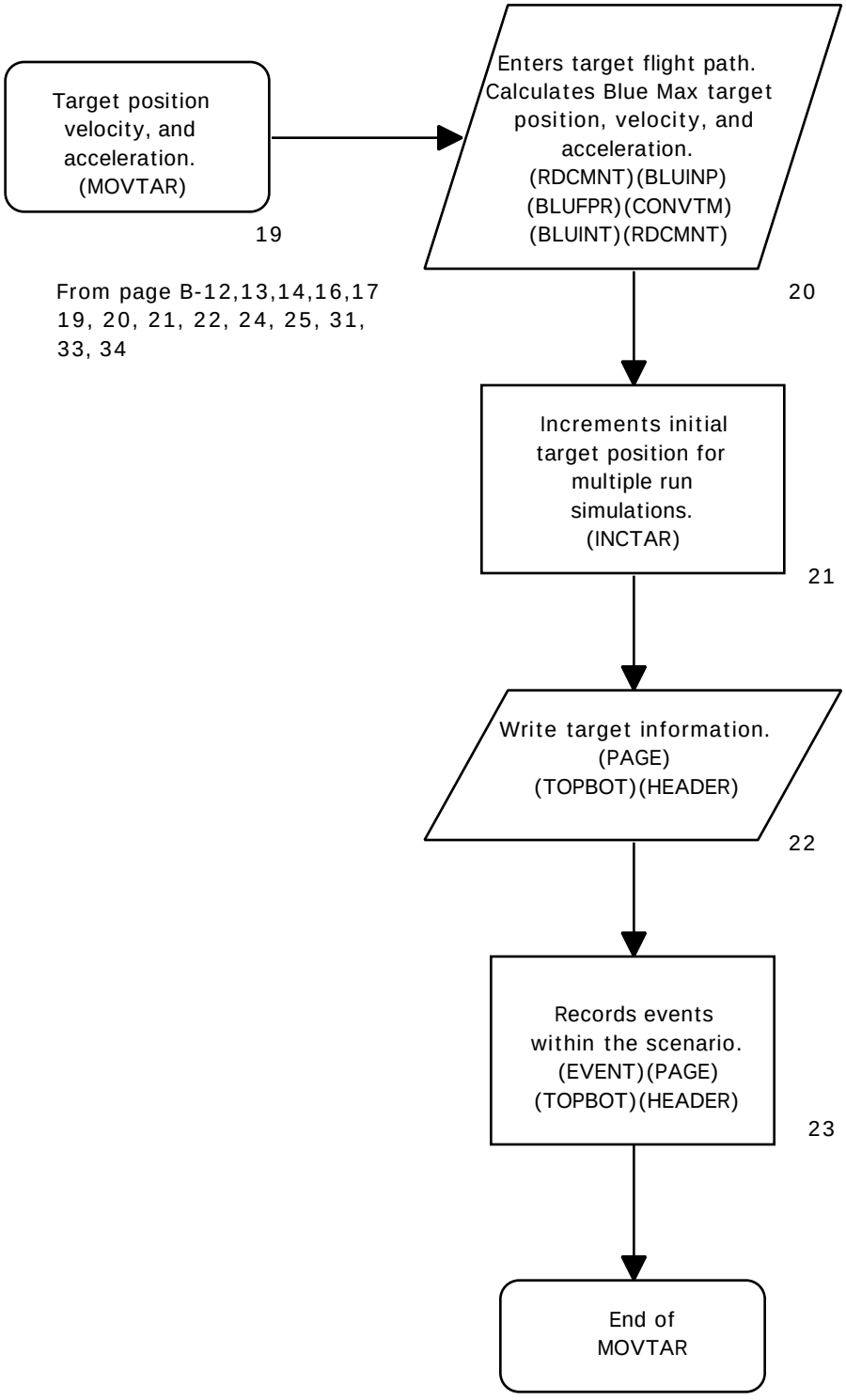

from page B-13
Search Mode
8
Sector search pattern. (SRCH1)
Page B-19
Circular search pattern. (SRCH2)
Page B-20
Antenna cued in target direction. (PERCUE)
Page B-21
Creates files of data for plots. (WRTMUL)
9
Computes target position,velocity, and acceleration. (MOVTAR)
Page B-18
Determines Page Layout and Prints Classification and Header. (PAGE)(TOPBOT)(HEADER)
10
Controls weapon system in autotrack mode. (ENGAGE)
Page B-22
Prints final report. (REPORT)(PAGE) (ROTBOT)(HEADER)
11
End of AAASIM

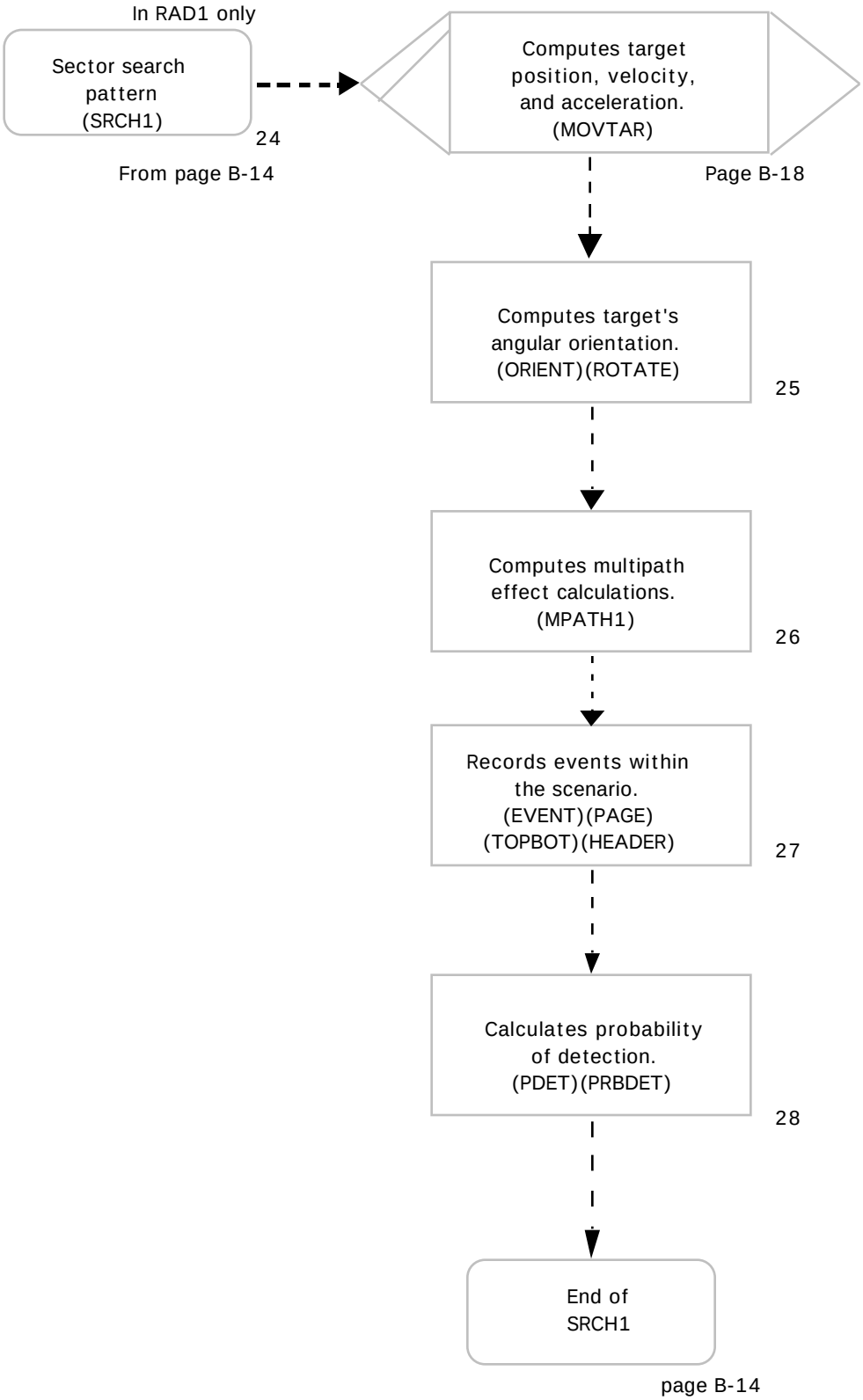

Weapon &
Scenario Inputs
(INPUT1)
12
1) Comments on run
2) Simulation type
3) When to end simulation
4) Antenna search pattern
5)Acquisition radar model
6) Operator reaction time for
detection-to-autotrack.
7) Type of target tracking
(RDCMNT)(PAGE)
(TOPBOT)(HEADER)
13
Checks tracking type is
realistic and prints out
any warning messages.
(CHKTRK)
(RDCMNT)(PAGE)
(TOPBOT)(HEADER)
14
End of
INPUT1

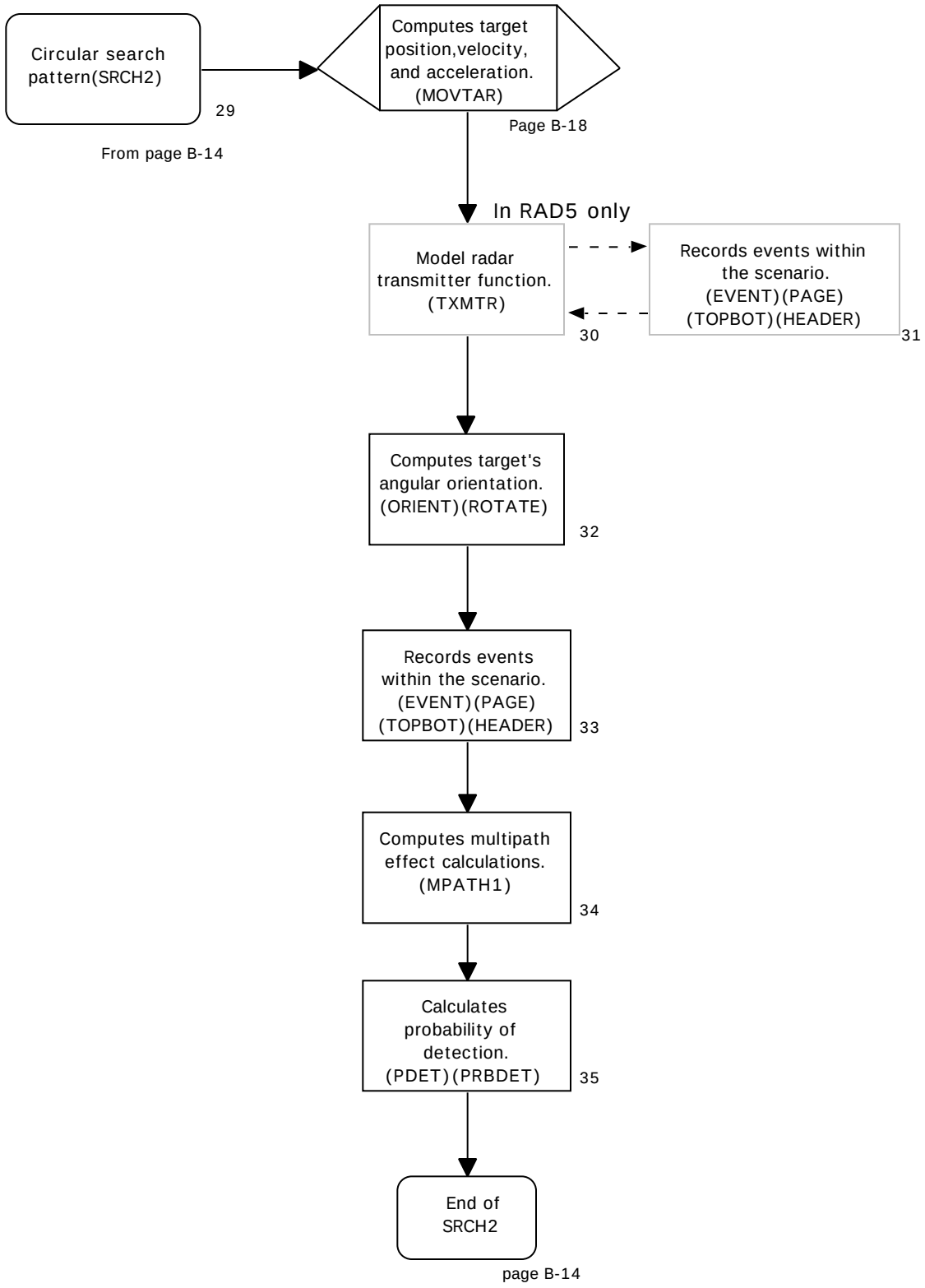
Page B-12

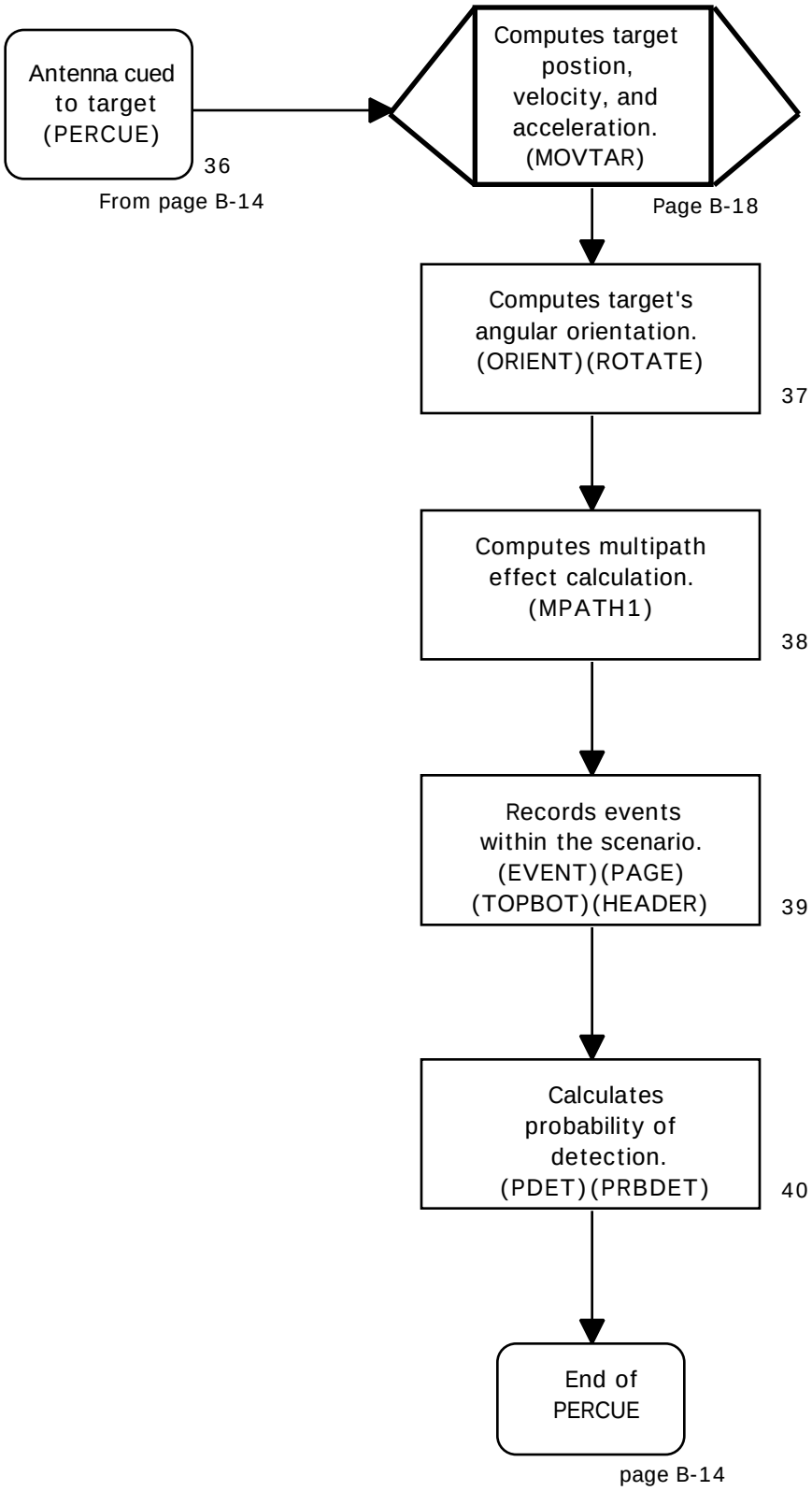

Weapon & Scenario Inputs (INPUT2)
15
from page B-12
8) MTI Switch
9) Operator reaction time for reacquisition of target
10) Disable guns
11) Override # of gun barrels in system
12) Override system fire rate
13) Select firing burst lengths and rest period
14) Override # of shells in magazine
15) Course heading and speed for moving weapon system
16) Scale RCS function
17) Number of target vulnerable areas (VARREAD)
18) Scale and view target areas
19) Initialize target flight path (MOVTAR) page B-18
20) Clutter/multipath model type
21) Hill in simulation?
22) Jamming in simulation (INPJAM)
23) Printout options
24) Create graphics file
25) Tracking error constants
26) Override order of fire control computer
(RDCMNT)(PAGE)(TOPBOT)(HEADER)
16
End of INPUT2
page B-12

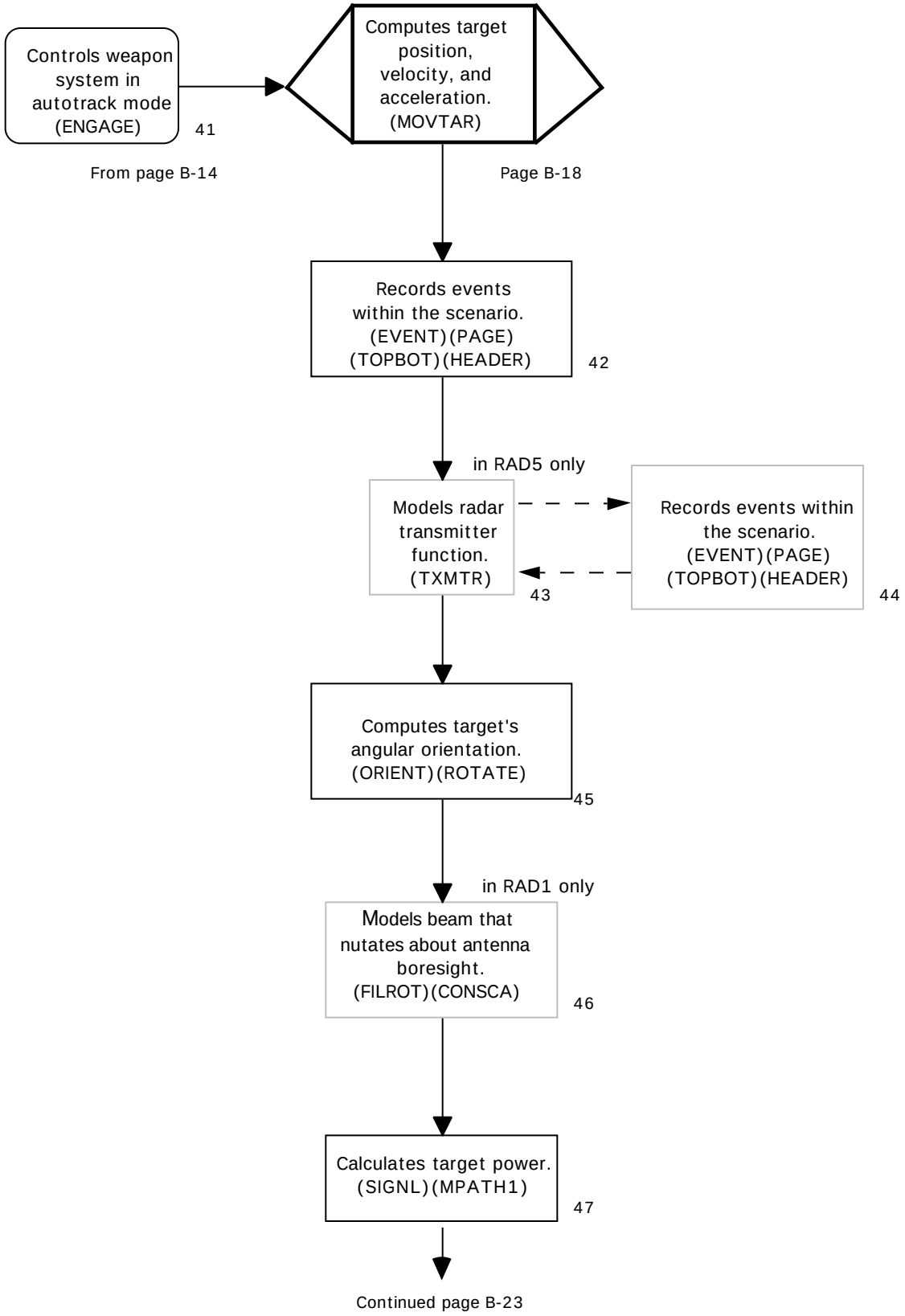

"Detection Range" Data files for plotting (DETRNG)
17
From page B-12
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Write target altitude, range, and elevation. (PAGE) (TOPBOT)(HEADER)
18
End of DETRNG
page B-12

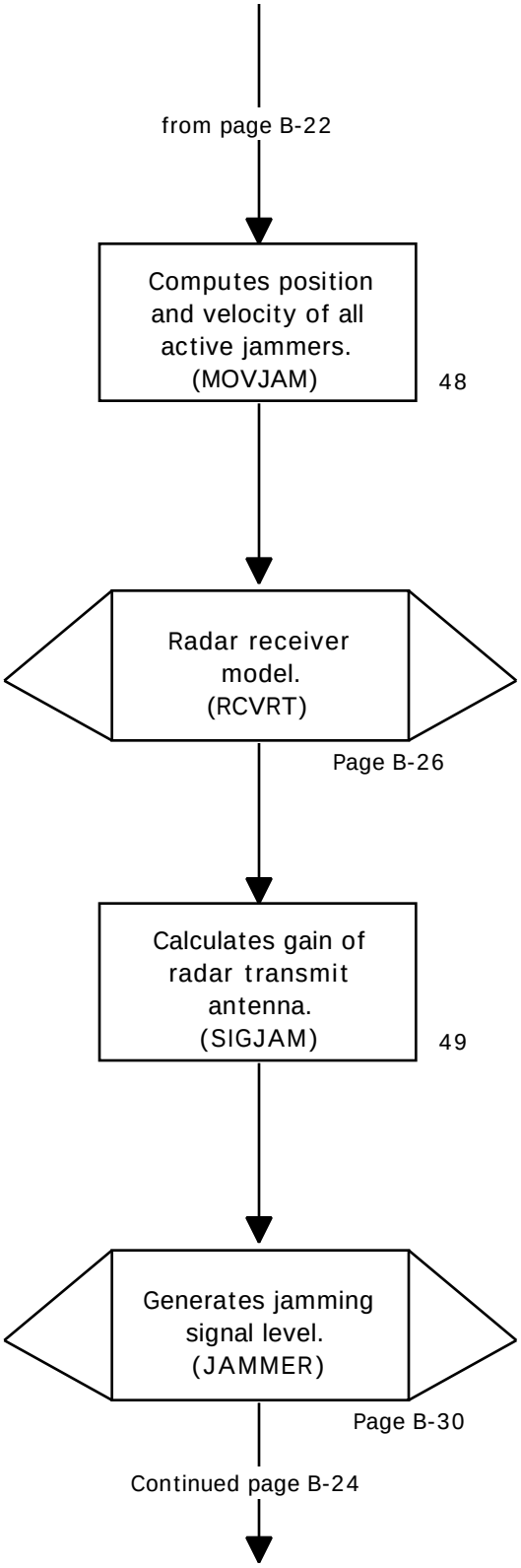

Target position velocity, and acceleration. (MOVTAR)
19
Enters target flight path. Calculates Blue Max target position, velocity, and acceleration. (RDCMNT)(BLUINP) (BLUFPR)(CONVTM) (BLUINT)(RDCMNT)
20
From page B-12,13,14,16,17 19, 20, 21, 22, 24, 25, 31, 33, 34
Increments initial target position for multiple run simulations. (INCTAR)
21
Write target information. (PAGE) (TOPBOT)(HEADER)
22
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
23
End of MOVTAR
page B-12,13,14,16,17 19, 20, 21, 22, 24, 25, 31, 33, 34

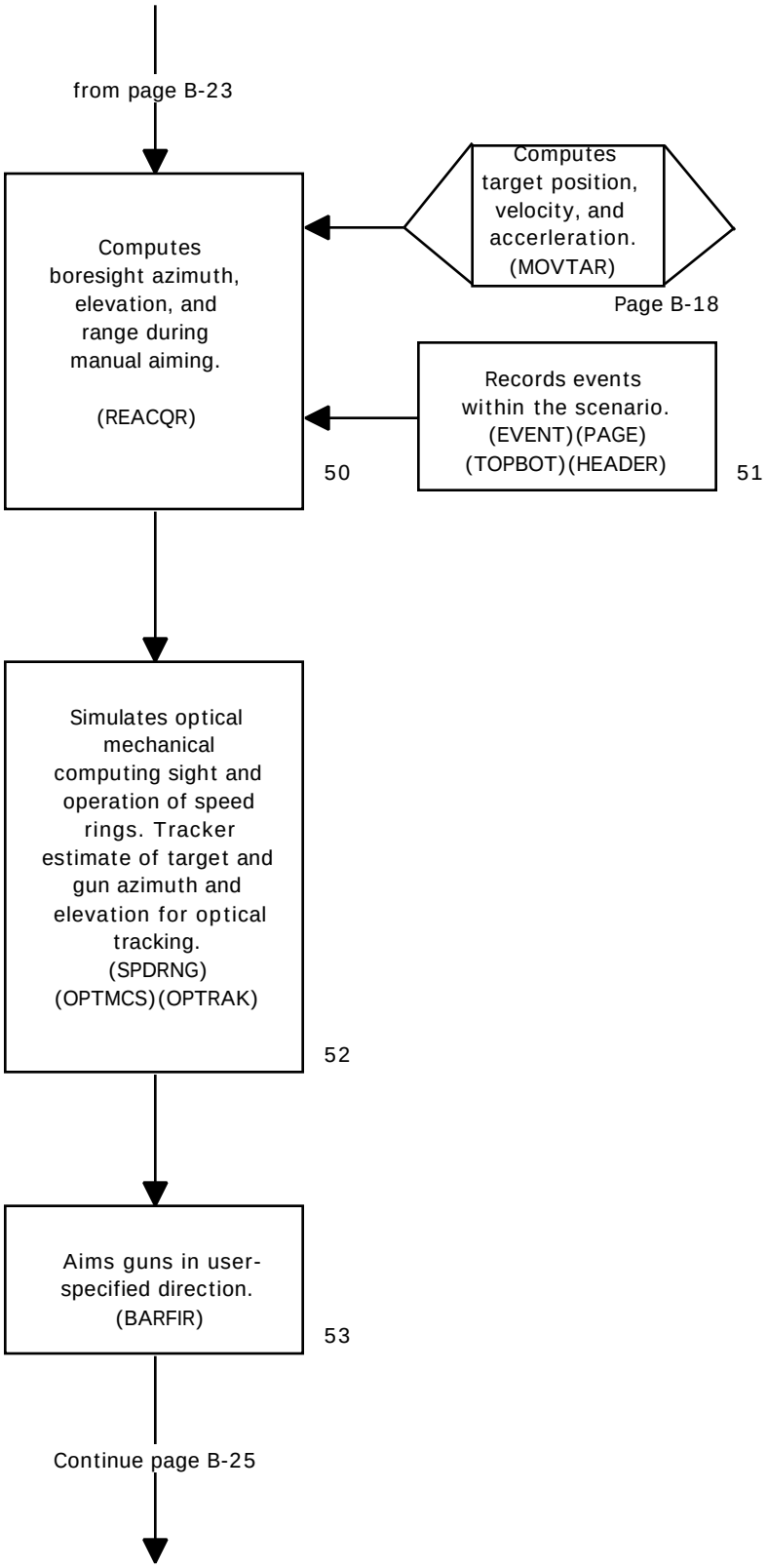

In RAD1 only
Sector search pattern (SRCH1)
24
From page B-14
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Computes target's angular orientation. (ORIENT)(ROTATE)
25
Computes multipath effect calculations. (MPATH1)
26
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
27
Calculates probability of detection. (PDET)(PRBDET)
28
End of SRCH1
page B-14

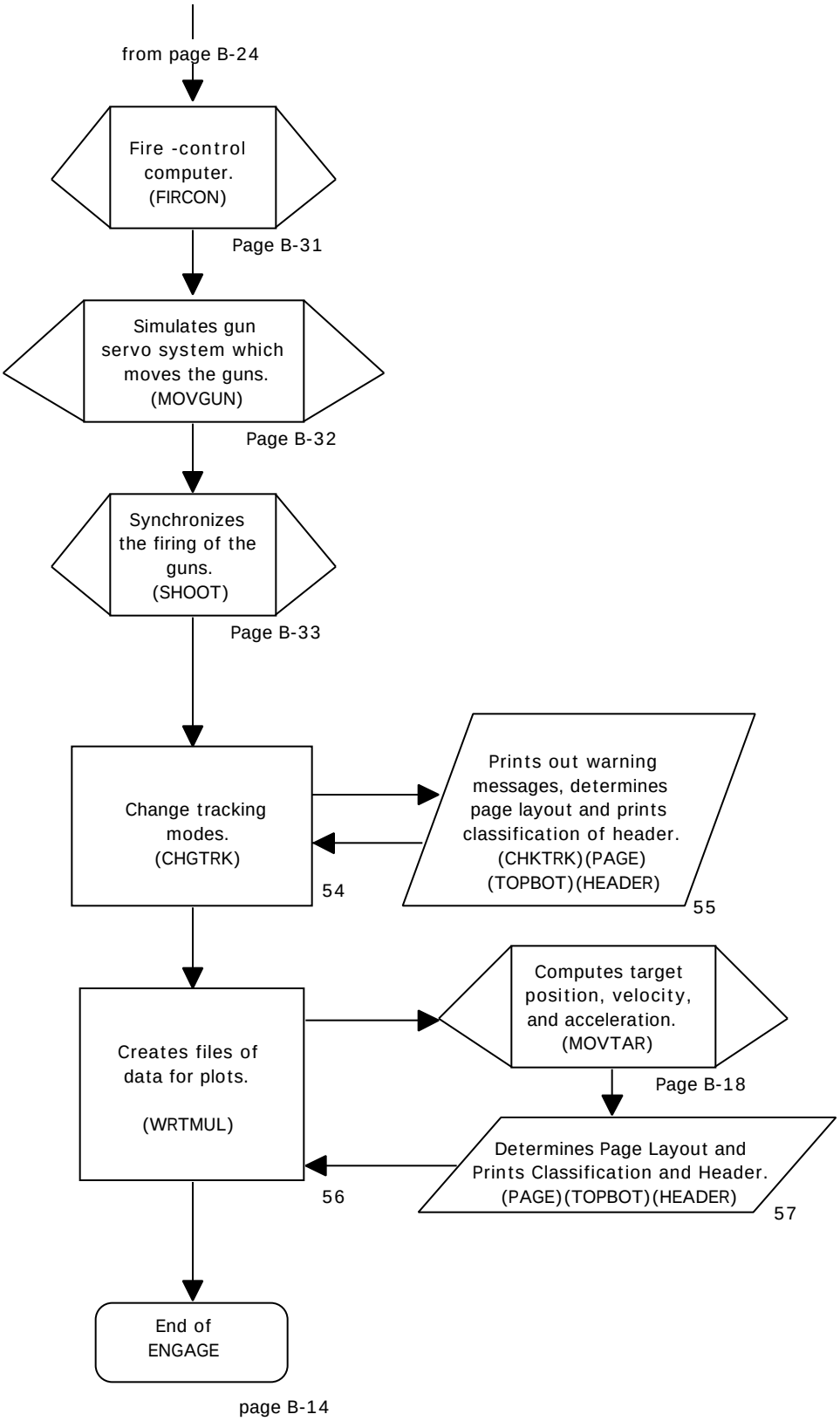

Circular search pattern(SRCH2)
29
From page B-14
Computes target position,velocity, and acceleration. (MOVTAR)
Page B-18
In RAD5 only
Model radar transmitter function. (TXMTR)
30
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
31
Computes target's angular orientation. (ORIENT)(ROTATE)
32
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
33
Computes multipath effect calculations. (MPATH1)
34
Calculates probability of detection. (PDET)(PRBDET)
35
End of SRCH2
page B-14

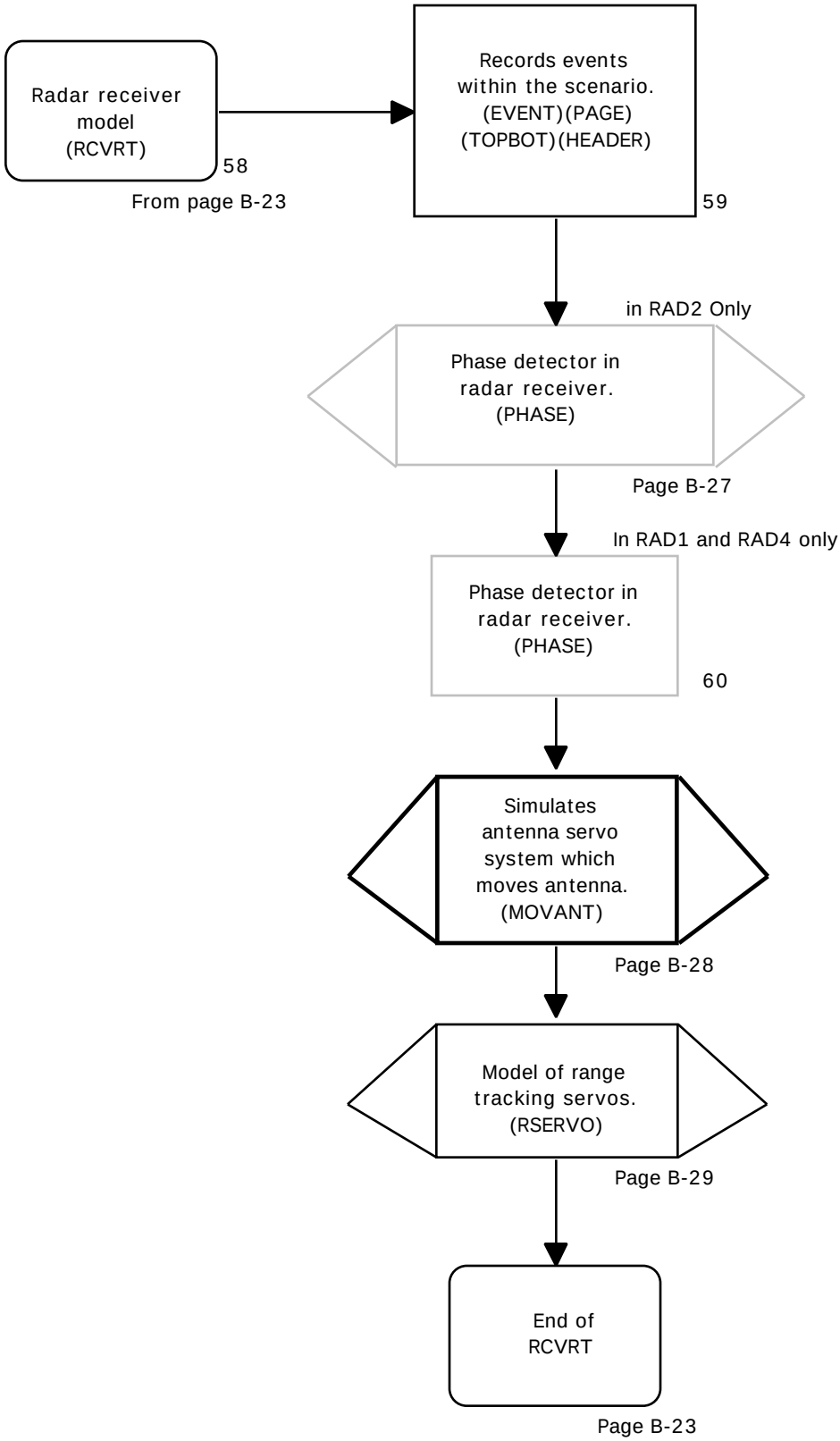

Antenna cued to target (PERCUE)
36
From page B-14
Computes target postion, velocity, and acceleration. (MOVTAR)
Page B-18
Computes target's angular orientation. (ORIENT)(ROTATE)
37
Computes multipath effect calculation. (MPATH1)
38
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
39
Calculates probability of detection. (PDET)(PRBDET)
40
End of PERCUE
page B-14

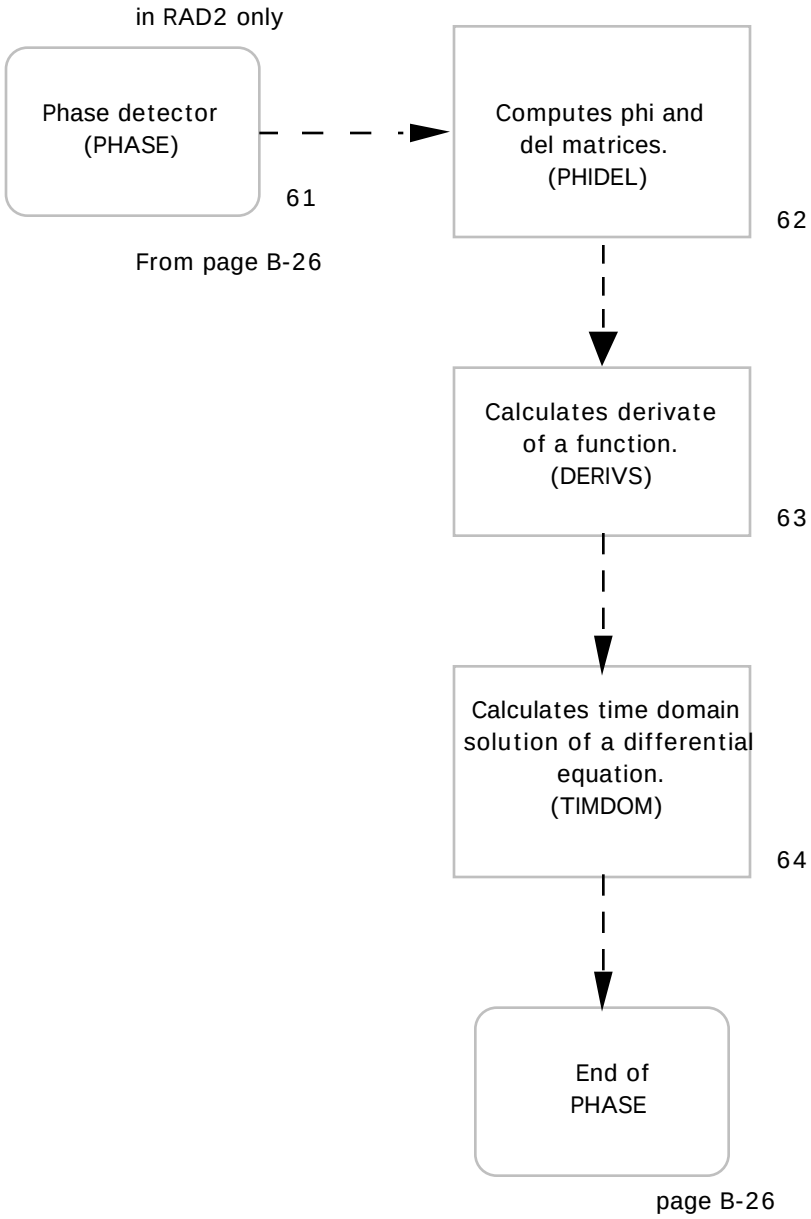

Controls weapon system in autotrack mode (ENGAGE)
41
From page B-14
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
42
in RAD5 only
Models radar transmitter function. (TXMTR)
43
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
44
Computes target's angular orientation. (ORIENT)(ROTATE)
45
in RAD1 only
Models beam that nutates about antenna boresight. (FILROT)(CONSCA)
46
Calculates target power. (SIGNL)(MPATH1)
47
Continued page B-23

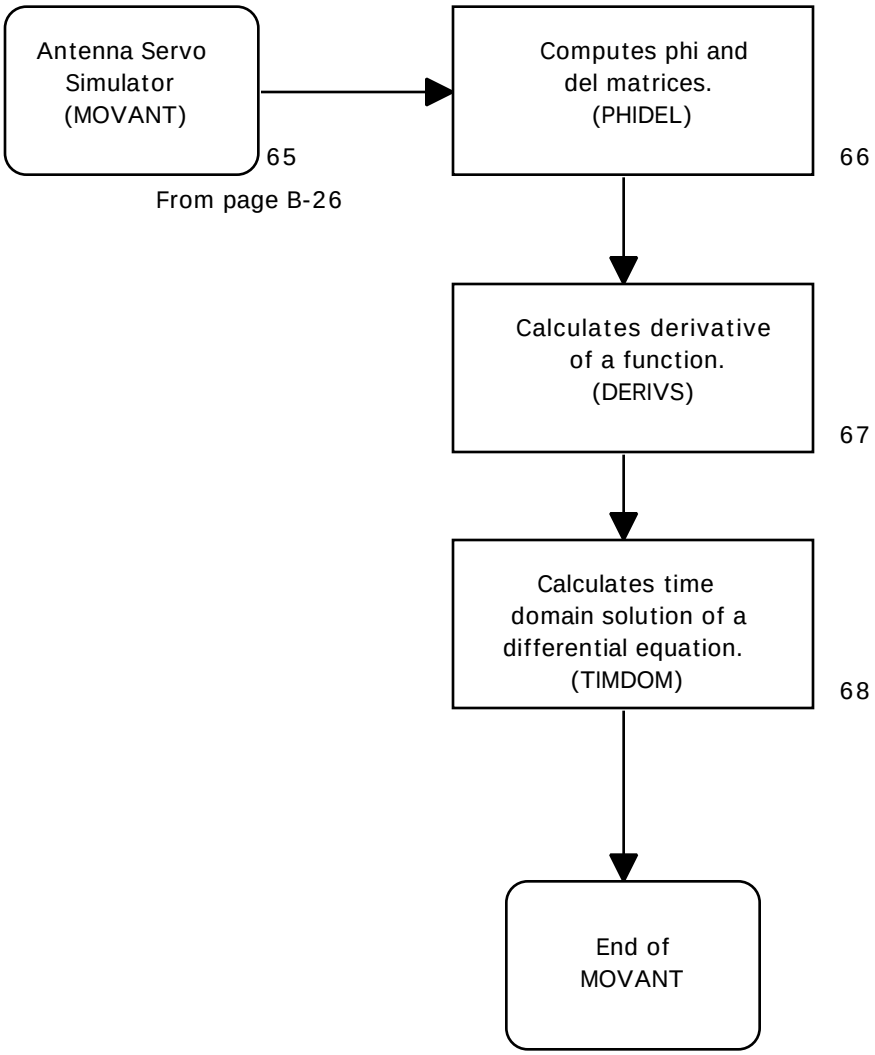

from page B-22
Computes position and velocity of all active jammers. (MOVJAM)
48
Radar receiver model. (RCVRT)
Page B-26
Calculates gain of radar transmit antenna. (SIGJAM)
49
Generates jamming signal level. (JAMMER)
Page B-30
Continued page B-24

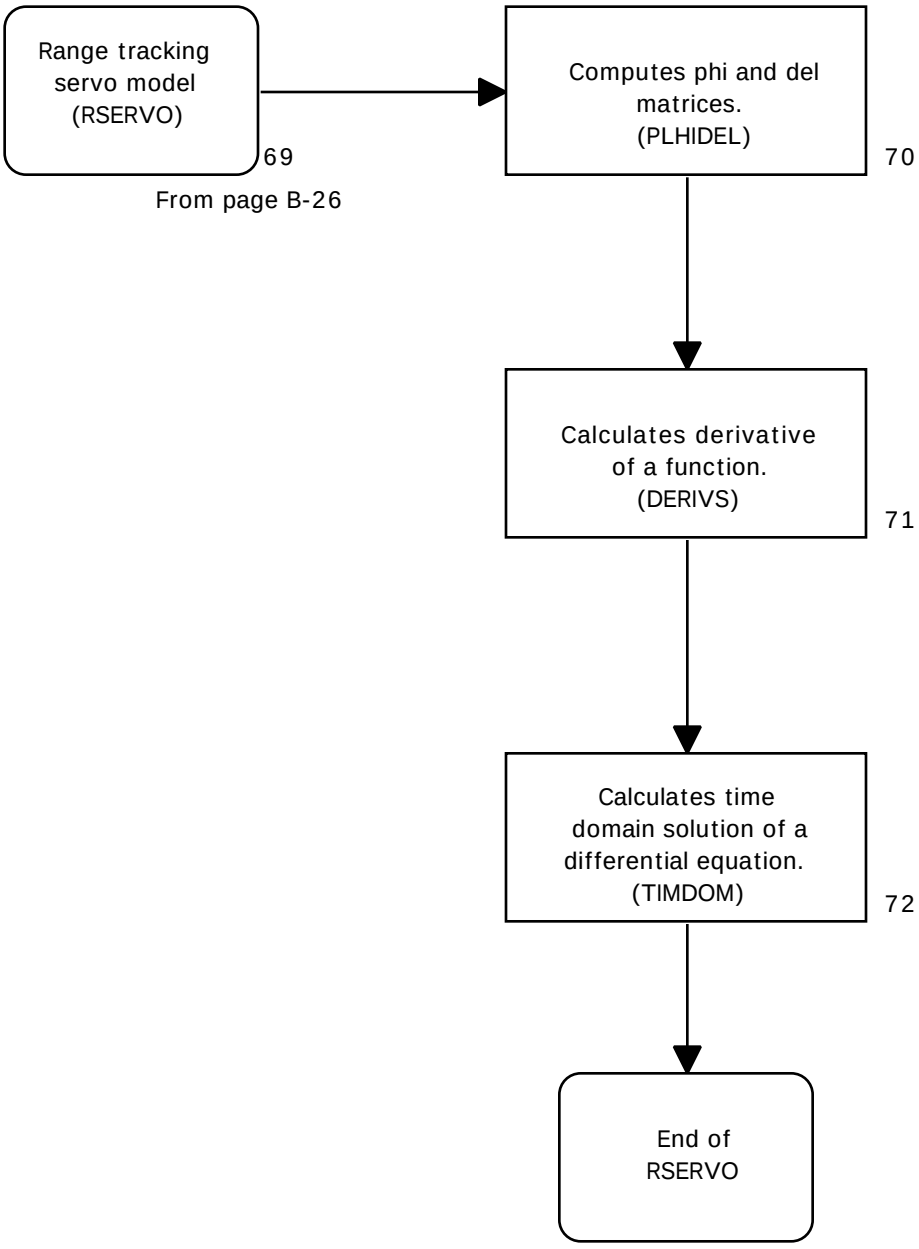

from page B-23
Computes target position, velocity, and accerleration. (MOVTAR)
Page B-18
Computes boresight azimuth, elevation, and range during manual aiming.
(REACQR)
50
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
51
Simulates optical mechanical computing sight and operation of speed rings. Tracker estimate of target and gun azimuth and elevation for optical tracking. (SPDRNG) (OPTMCS)(OPTRAK)
52
Aims guns in user-specified direction. (BARFIR)
53
Continue page B-25

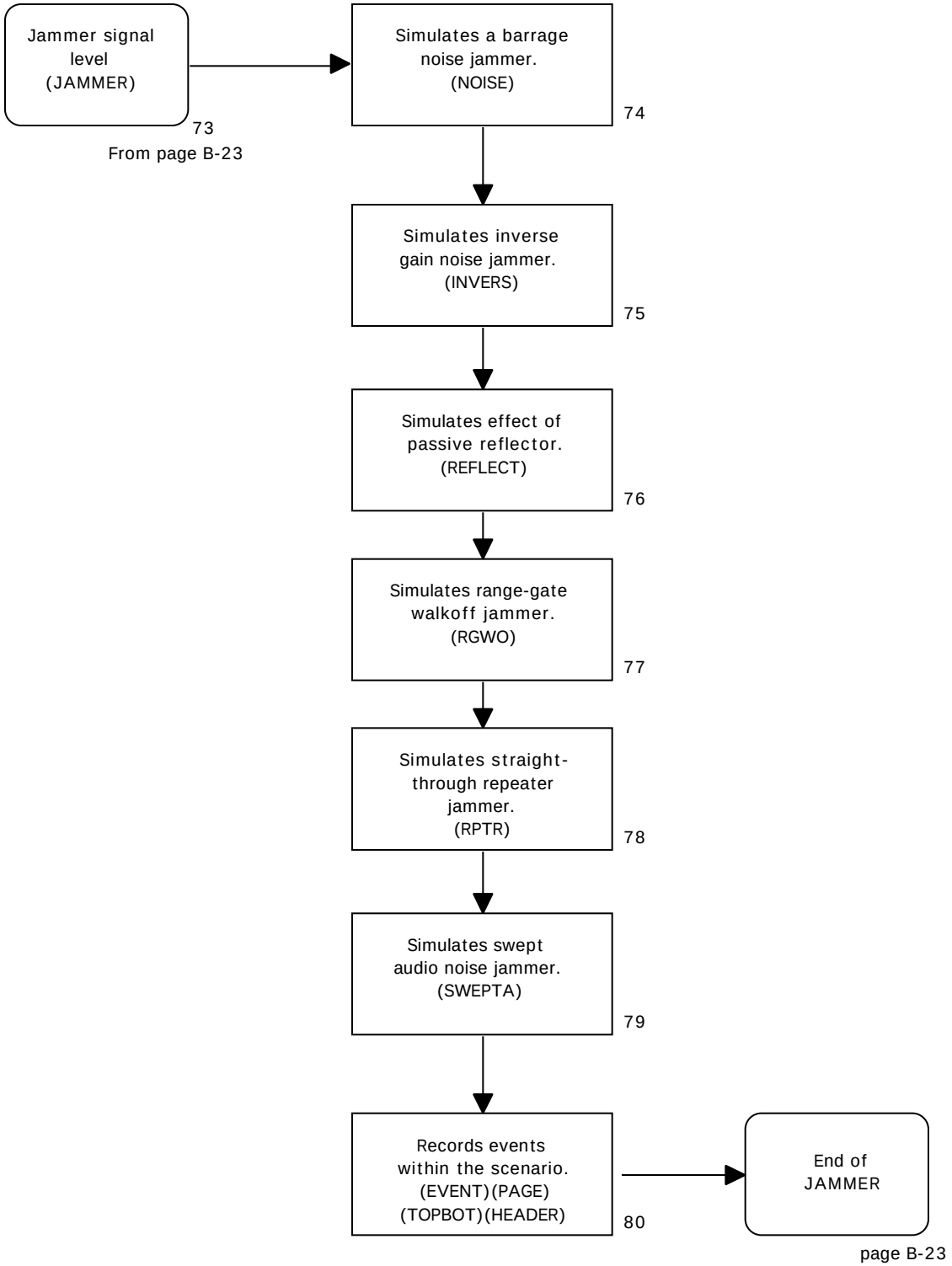

from page B-24
Fire -control computer. (FIRCON)
Page B-31
Simulates gun servo system which moves the guns. (MOVGUN)
Page B-32
Synchronizes the firing of the guns. (SHOOT)
Page B-33
Change tracking modes. (CHGTRK)
54
Prints out warning messages, determines page layout and prints classification of header. (CHKTRK)(PAGE) (TOPBOT)(HEADER)
55
Creates files of data for plots. (WRTMUL)
56
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Determines Page Layout and Prints Classification and Header. (PAGE)(TOPBOT)(HEADER)
57
End of ENGAGE
page B-14

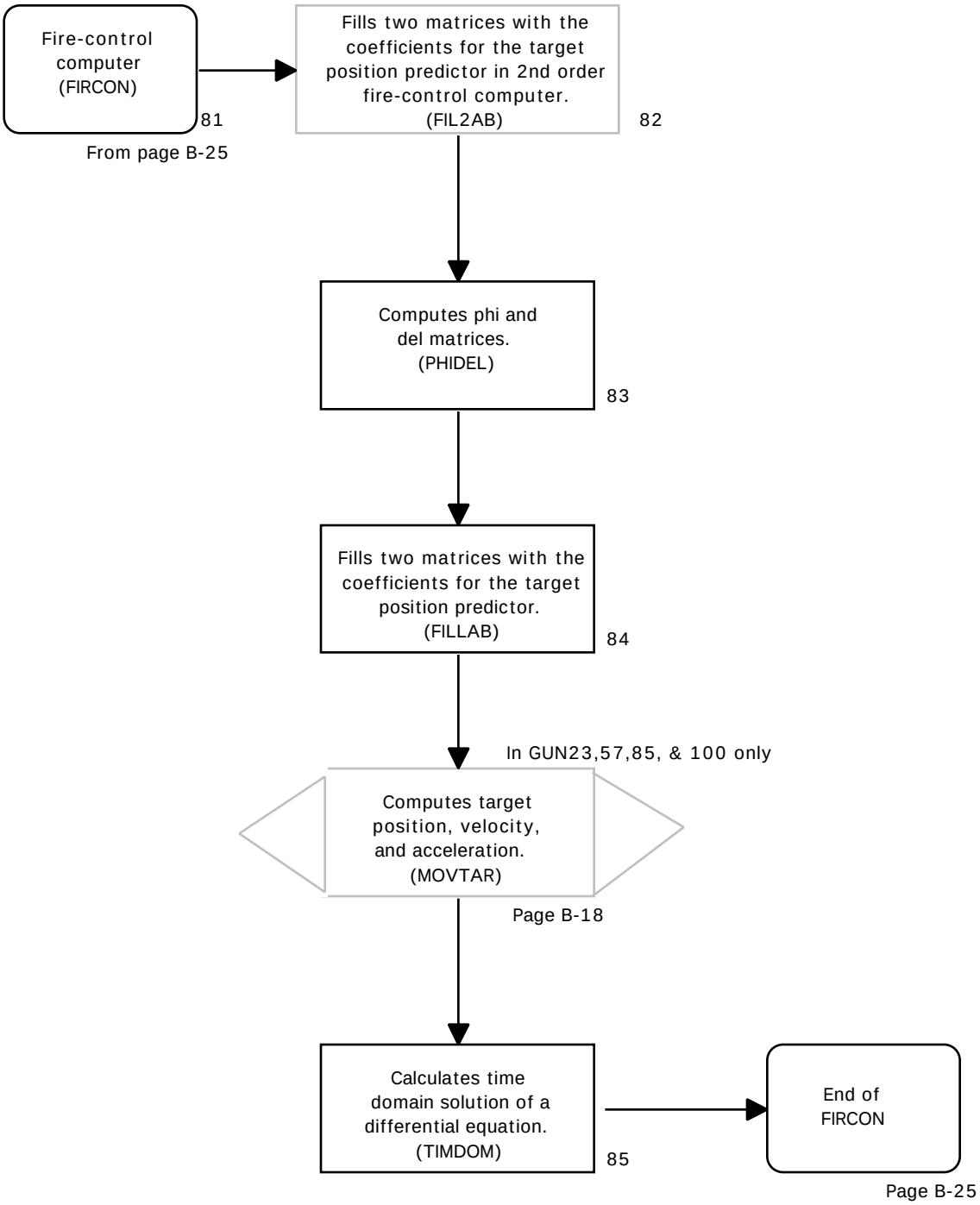

Radar receiver model (RCVRT)
58
From page B-23
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
59
in RAD2 Only
Phase detector in radar receiver. (PHASE)
Page B-27
In RAD1 and RAD4 only
Phase detector in radar receiver. (PHASE)
60
Simulates antenna servo system which moves antenna. (MOVANT)
Page B-28
Model of range tracking servos. (RSERVO)
Page B-29
End of RCVRT
Page B-23

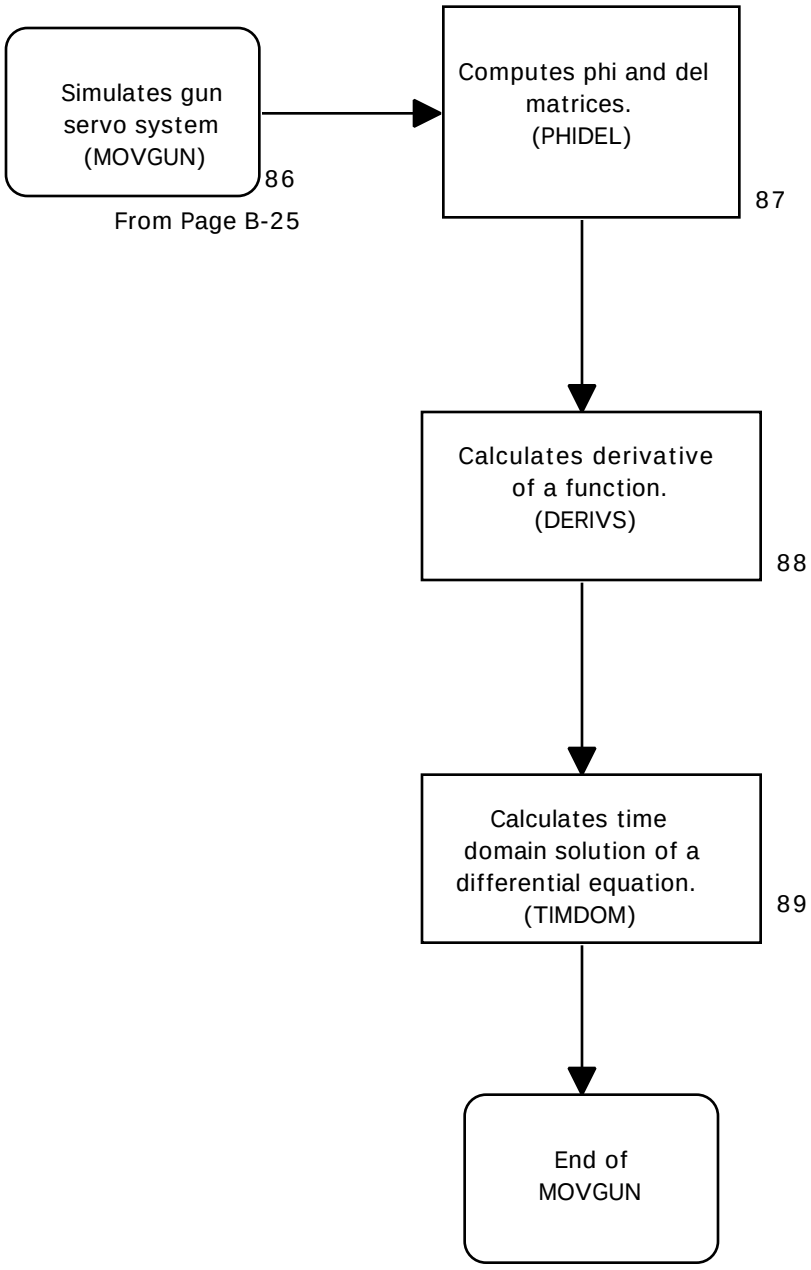

in RAD2 only
Phase detector
(PHASE)
61
Computes phi and
del matrices.
(PHIDEL)
62
From page B-26
Calculates derivate
of a function.
(DERIVS)
63
Calculates time domain
solution of a differential
equation.
(TIMDOM)
64
End of
PHASE
page B-26

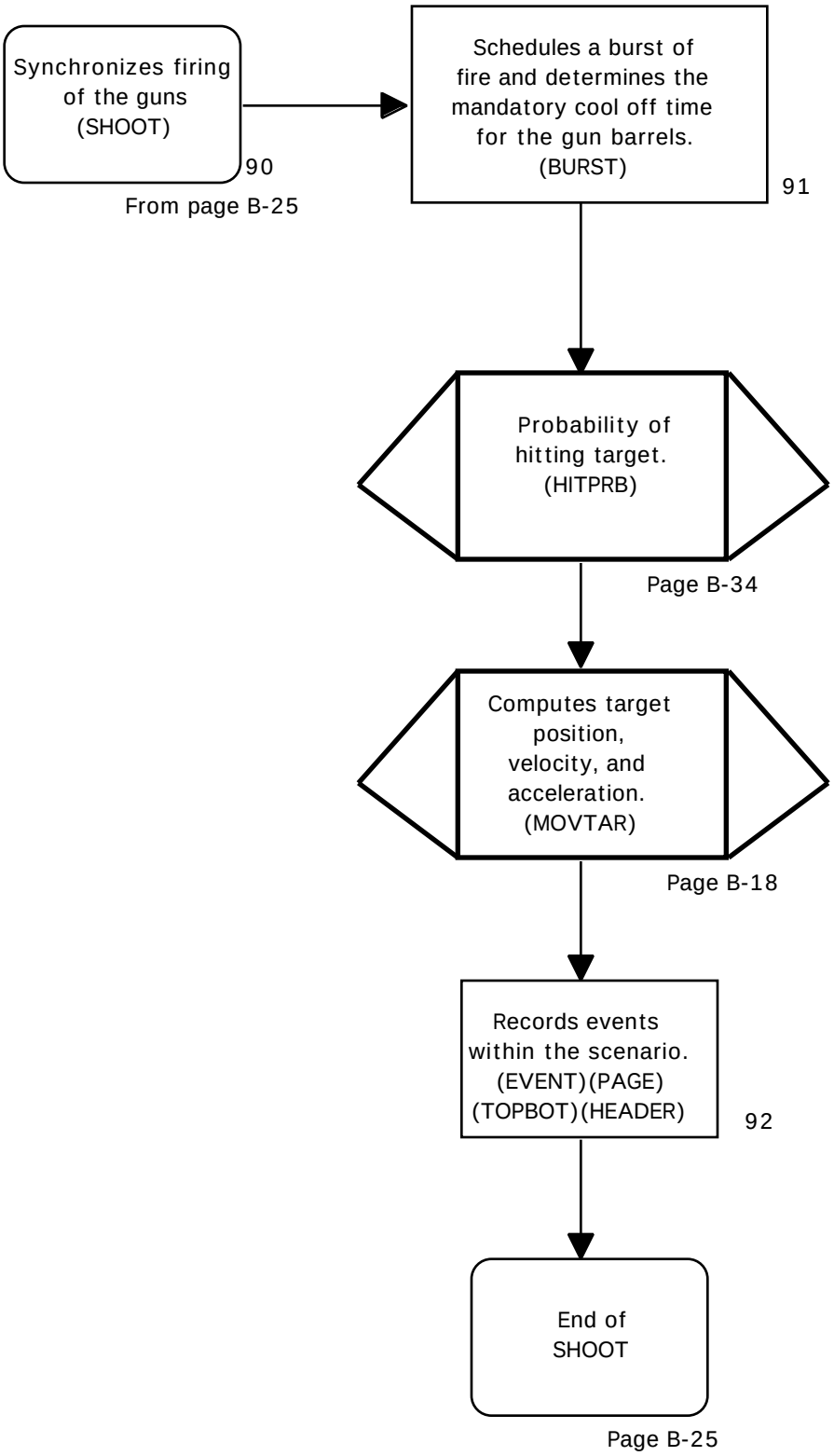

Antenna Servo Simulator (MOVANT)
65
From page B-26
Computes phi and del matrices. (PHIDEL)
66
Calculates derivative of a function. (DERIVS)
67
Calculates time domain solution of a differential equation. (TIMDOM)
68
End of MOVANT
page B-26

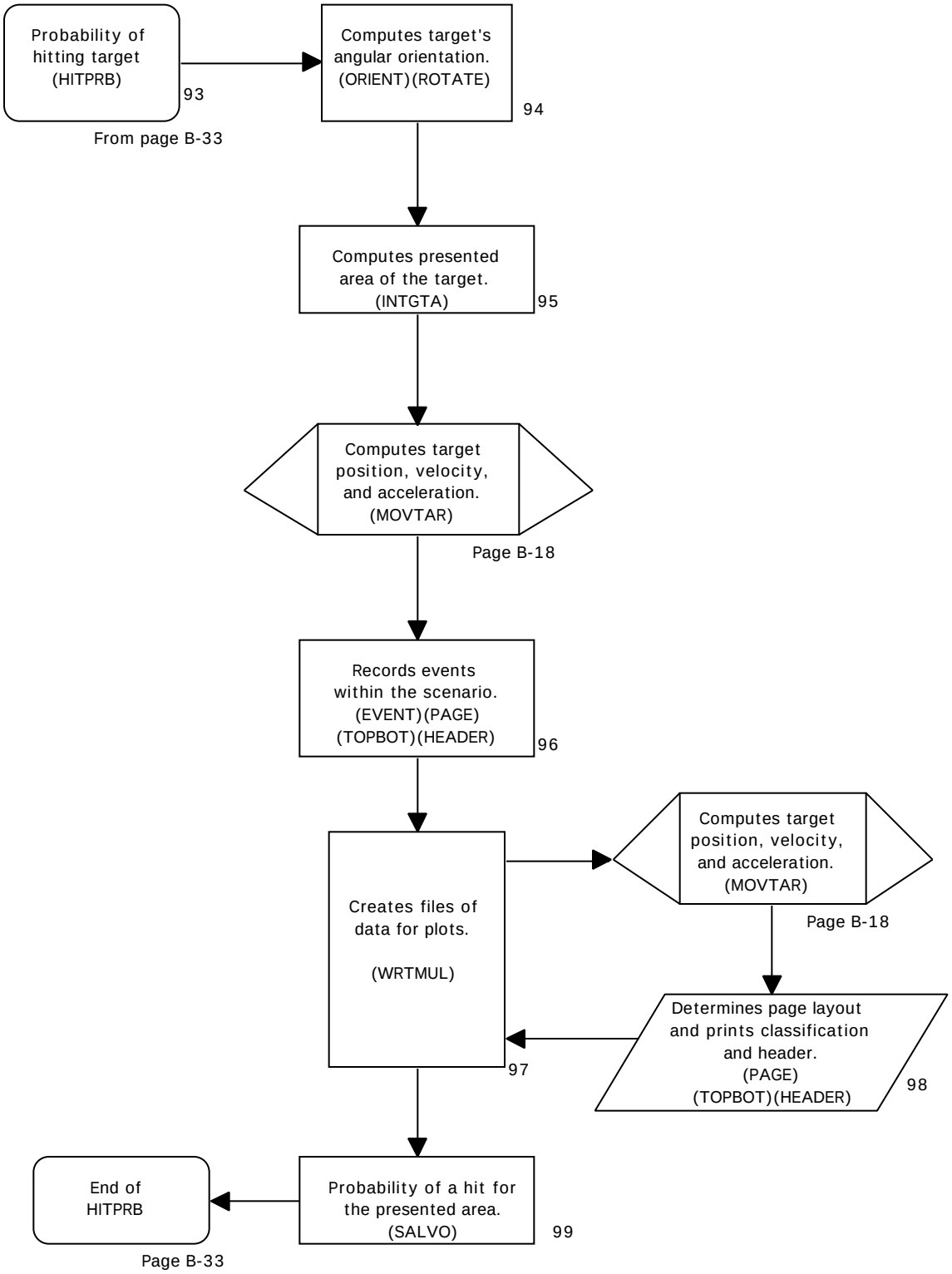

Range tracking servo model (RSERVO)
69
From page B-26
Computes phi and del matrices. (PLHIDEL)
70
Calculates derivative of a function. (DERIVS)
71
Calculates time domain solution of a differential equation. (TIMDOM)
72
End of RSERVO
Page B-26


Jammer signal level (JAMMER)
73
From page B-23
Simulates a barrage noise jammer. (NOISE)
74
Simulates inverse gain noise jammer. (INVERS)
75
Simulates effect of passive reflector. (REFLECT)
76
Simulates range-gate walkoff jammer. (RGWO)
77
Simulates straight-through repeater jammer. (RPTR)
78
Simulates swept audio noise jammer. (SWEPTA)
79
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
80
End of JAMMER
page B-23


Fire-control computer (FIRCON)
81
From page B-25
Fills two matrices with the coefficients for the target position predictor in 2nd order fire-control computer. (FIL2AB)
82
Computes phi and del matrices. (PHIDEL)
83
Fills two matrices with the coefficients for the target position predictor. (FILLAB)
84
In GUN23,57,85, & 100 only
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Calculates time domain solution of a differential equation. (TIMDOM)
85
End of FIRCON
Page B-25


Simulates gun servo system (MOVGUN)
86
From Page B-25
Computes phi and del matrices. (PHIDEL)
87
Calculates derivative of a function. (DERIVS)
88
Calculates time domain solution of a differential equation. (TIMDOM)
89
End of MOVGUN
Page B-25


Synchronizes firing of the guns (SHOOT)
90
From page B-25
Schedules a burst of fire and determines the mandatory cool off time for the gun barrels. (BURST)
91
Probability of hitting target. (HITPRB)
Page B-34
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
92
End of SHOOT
Page B-25


Probability of hitting target (HITPRB)
93
From page B-33
Computes target's angular orientation. (ORIENT)(ROTATE)
94
Computes presented area of the target. (INTGTA)
95
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Records events within the scenario. (EVENT)(PAGE) (TOPBOT)(HEADER)
96
Creates files of data for plots.
(WRTMUL)
97
Computes target position, velocity, and acceleration. (MOVTAR)
Page B-18
Determines page layout and prints classification and header. (PAGE) (TOPBOT)(HEADER)
98
Probability of a hit for the presented area. (SALVO)
99
End of HITPRB
Page B-33